Part II: Analysis and Forecast Modelling

Sally Chen

Contents

```r
library(fpp)
library(fpp2)
library(forecast)
library(GGally)
library(knitr)
```

```r
# Load cleaned data
full = read.csv("full.csv")
train = read.csv("train.csv")
test = read.csv("test.csv")
```

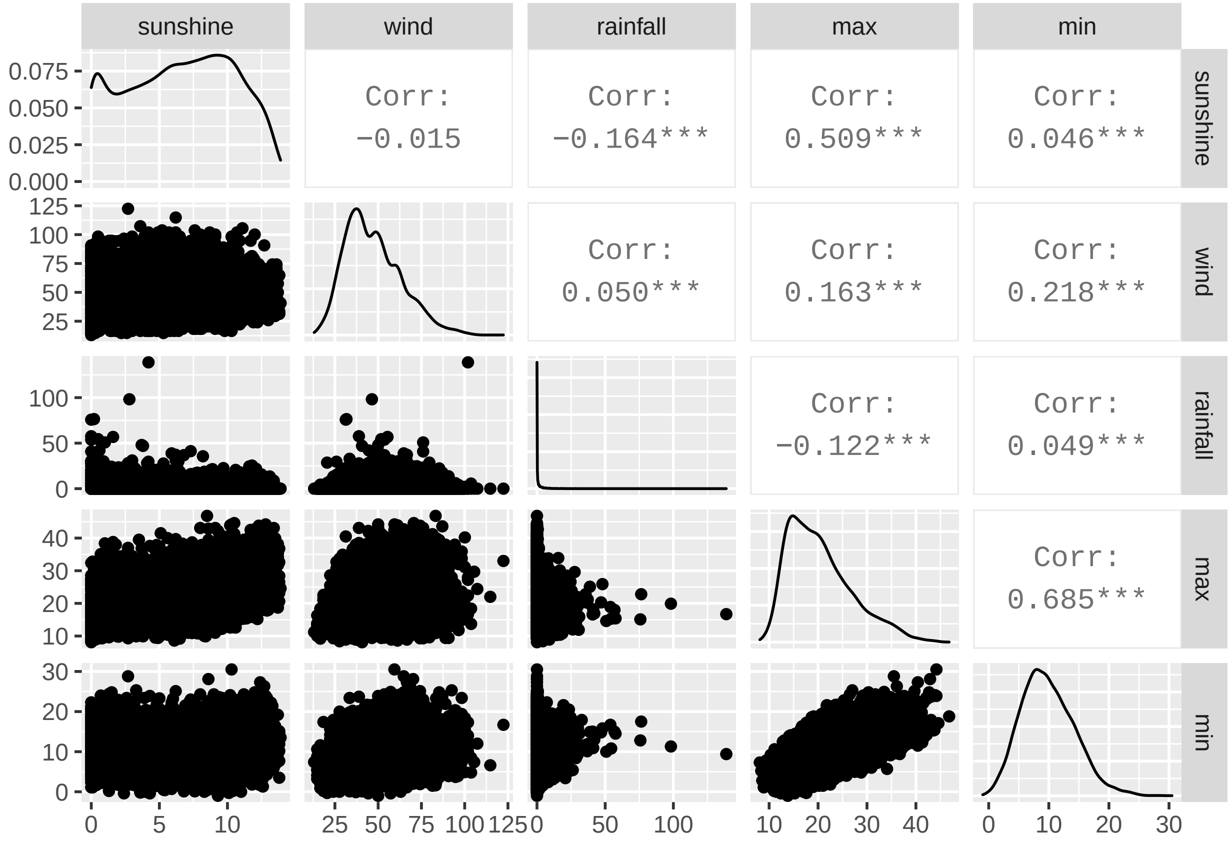
Defining Weather Elements as Time Series Objects

```r
ts_full = ts(full, frequency = 365.25, start = c(1999, 230))
ts_train = ts(train, frequency = 365.25, start = c(1999, 230))
ts_test = ts(test, frequency = 365.25, start = c(2016, 55))

ts_max = ts_train[, "max"]
```

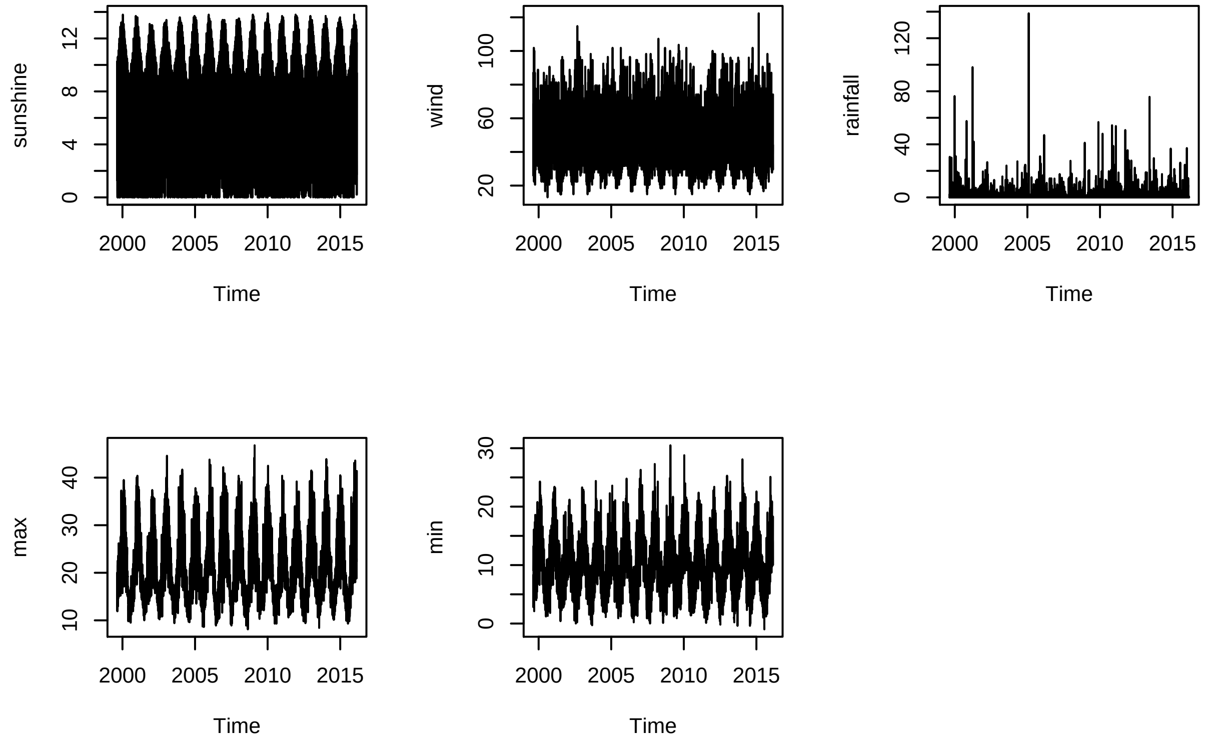
```
ts_min = ts_train[, "min"]
ts_sunshine = ts_train[, "sunshine"]
ts_wind = ts_train[, "wind"]
ts_rainfall = ts_train[, "rainfall"]
```

Train and Test Data Exploration

```
# Check variables correlations
GGally::ggpairs(train[, 1:5])
```

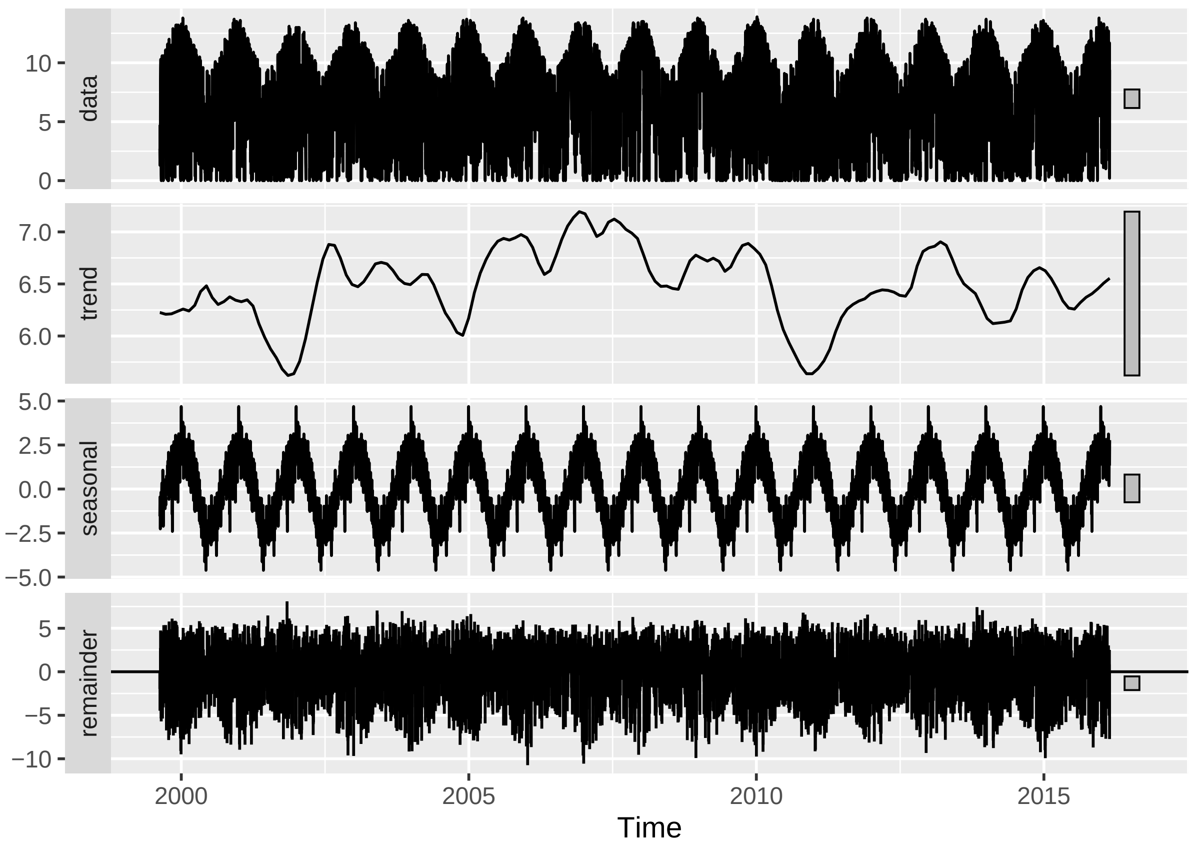


```
## Time series plots for weather elements
elements = colnames(ts_train)
par(mfrow = c(2, 3))
for (i in 1:5) {
    ts.plot(ts_train[, i], type = "l", ylab = elements[i])
}
```

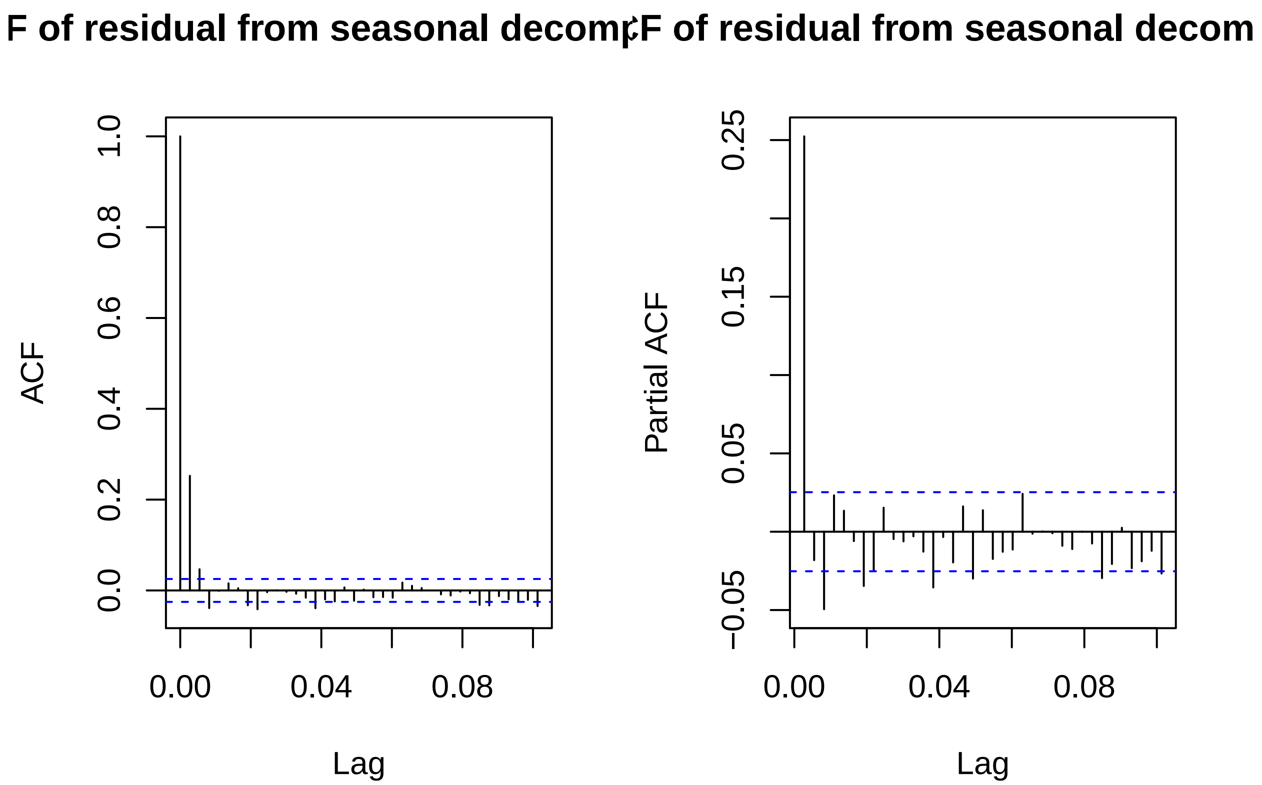


Inspect seasonal patterns from weather elements

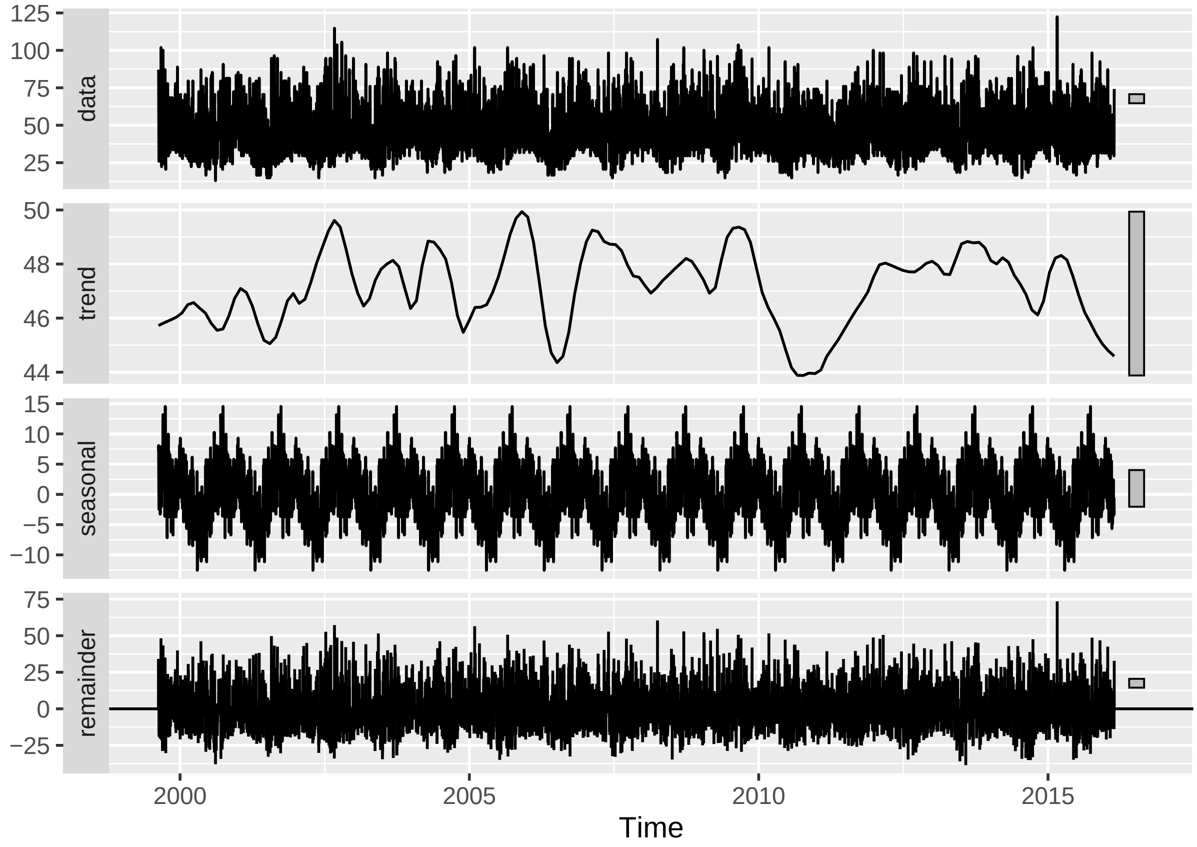
```r
# Sunshine -> seasonality, no trend
fitstl_sunshine = stl(ts_sunshine, t.window = 365.25, s.window = 365.25)
autoplot(fitstl_sunshine)
```



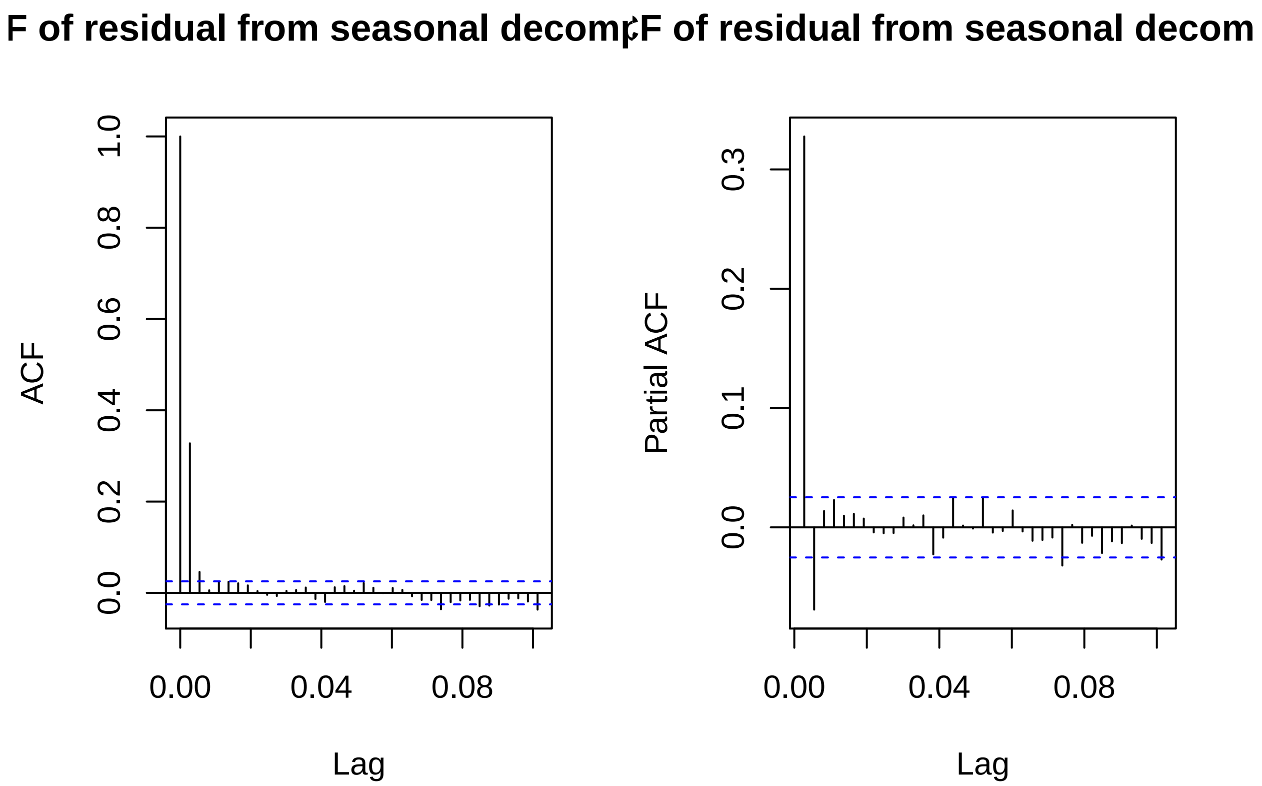
```
par(mfrow = c(1, 2))
acf(remainder(fitstl_sunshine), main = "ACF of residual from seasonal decomposition")
pacf(remainder(fitstl_sunshine), main = "PACF of residual from seasonal decomposition")
```



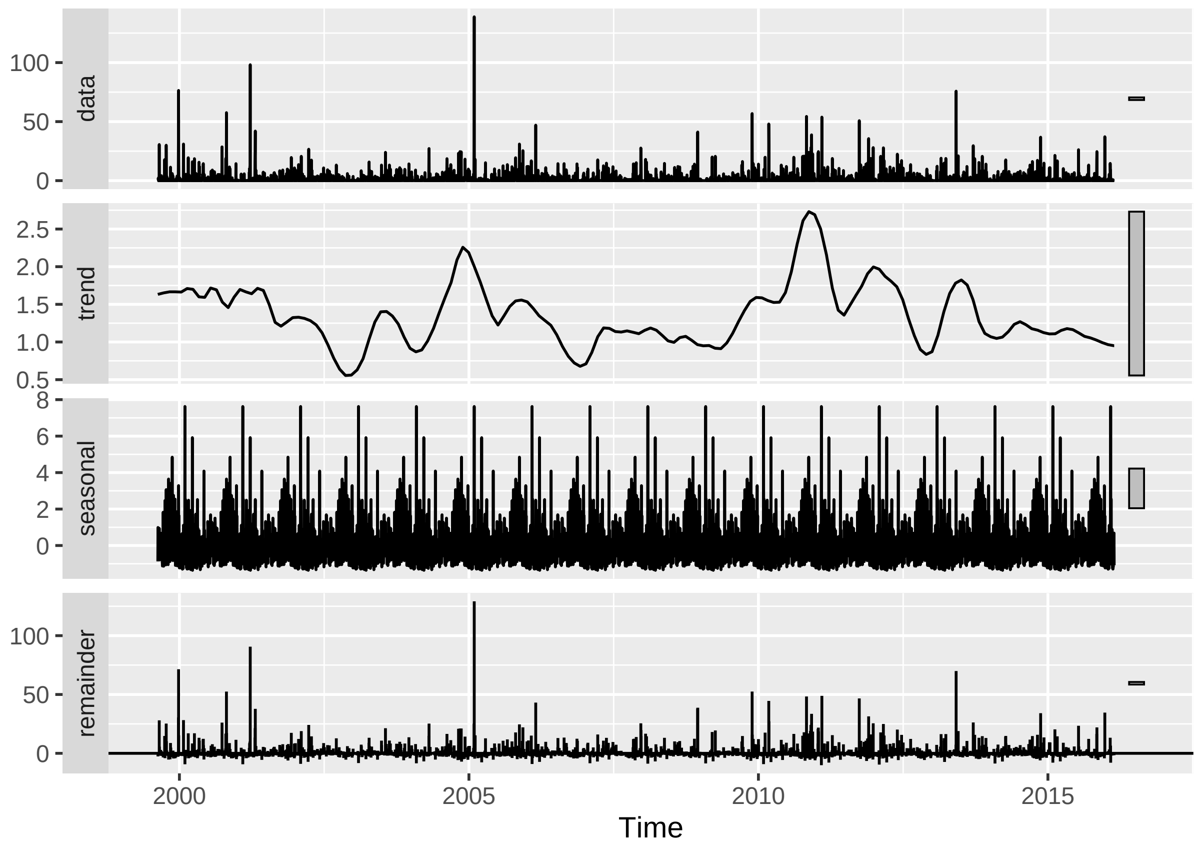
```
# wind -> seasonality, no trend
fitstl_wind = stl(ts_wind, t.window = 365.25, s.window = 365.25)
autoplot(fitstl_wind)
```



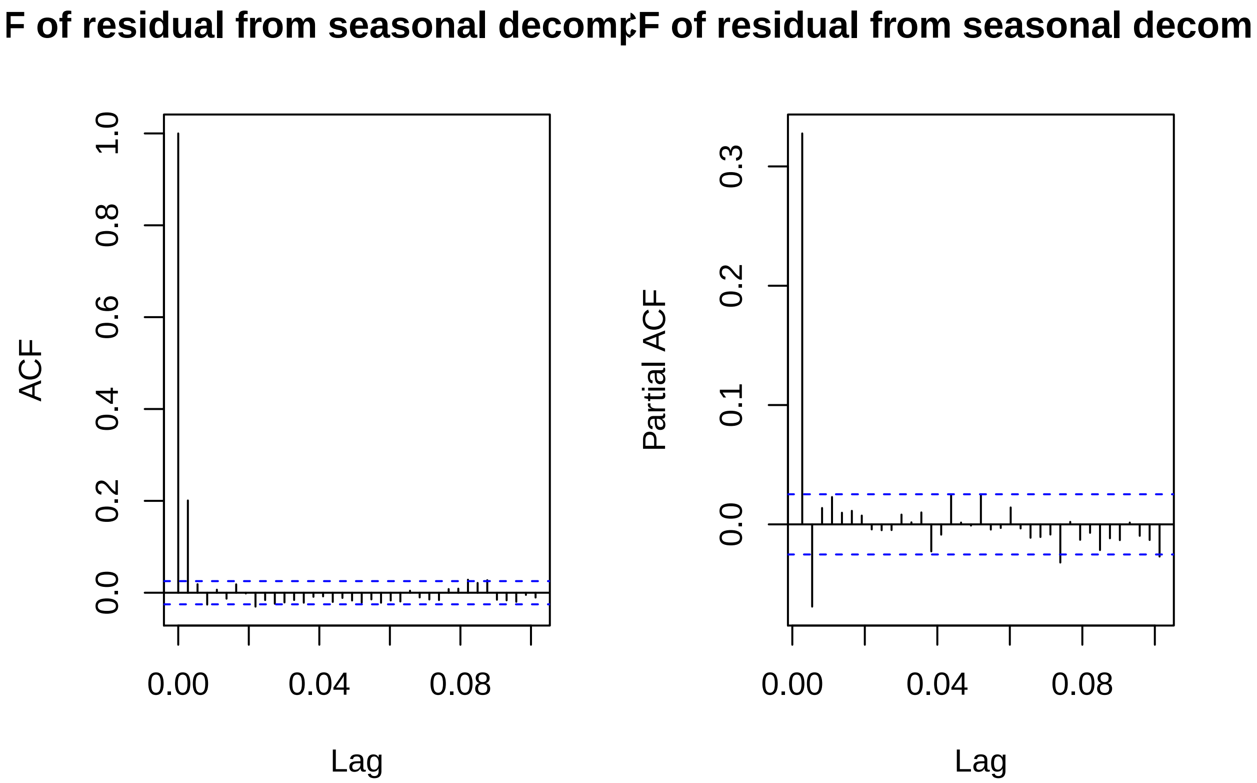
```
par(mfrow = c(1, 2))
acf(remainder(fitstl_wind), main = "ACF of residual from seasonal decomposition")
pacf(remainder(fitstl_wind), main = "PACF of residual from seasonal decomposition")
```



```r
# rainfall -> no seasonality, no trend
fitstl_rainfall = stl(ts_rainfall, t.window = 365.25, s.window = 365.25)
autoplot(fitstl_rainfall)
```

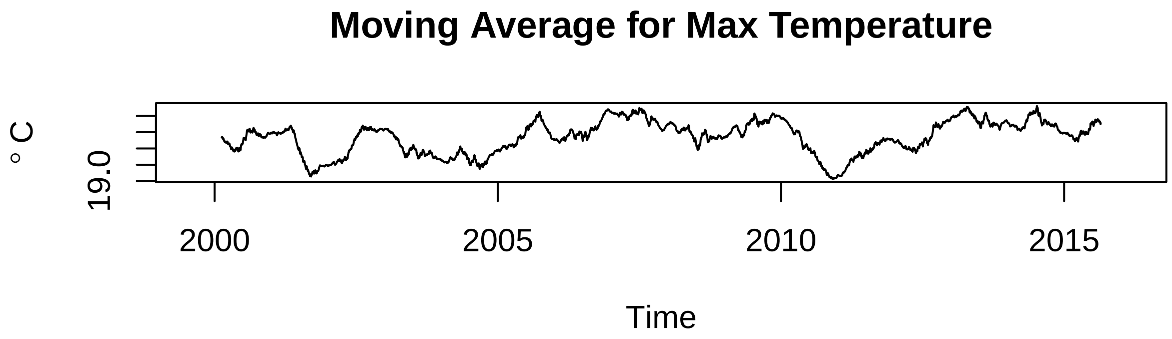


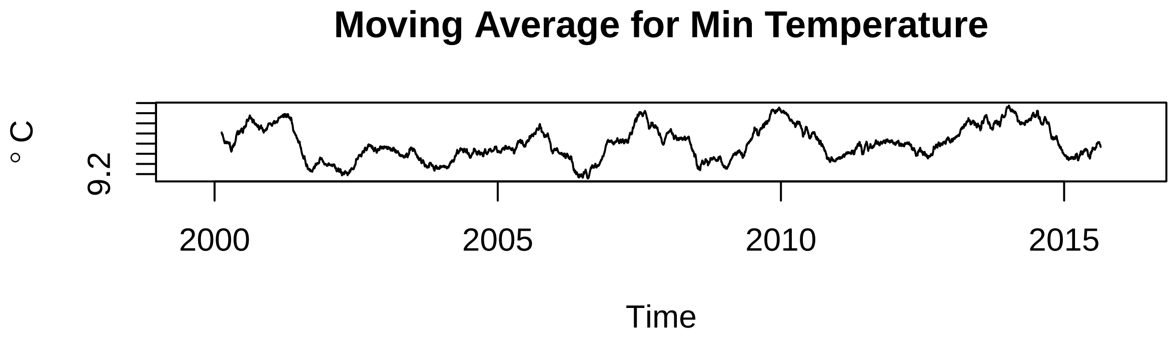
```r
par(mfrow = c(1, 2))
acf(remainder(fitstl_rainfall), main = "ACF of residual from seasonal decomposition")
pacf(remainder(fitstl_wind), main = "PACF of residual from seasonal decomposition")
```



Analyse trend in daily minimum and maximum temperature

```
## use moving average
par(mfrow = c(2, 1))
plot(ma(ts_max, order = 365), main = "Moving Average for Max Temperature", ylab =
↪  expression(degree ~
    C))
plot(ma(ts_min, order = 365), main = "Moving Average for Min Temperature", ylab =
↪  expression(degree ~
    C))
```





Building models to predict maximum temperature

1. Linear model with a Fourier term to capture seasonality

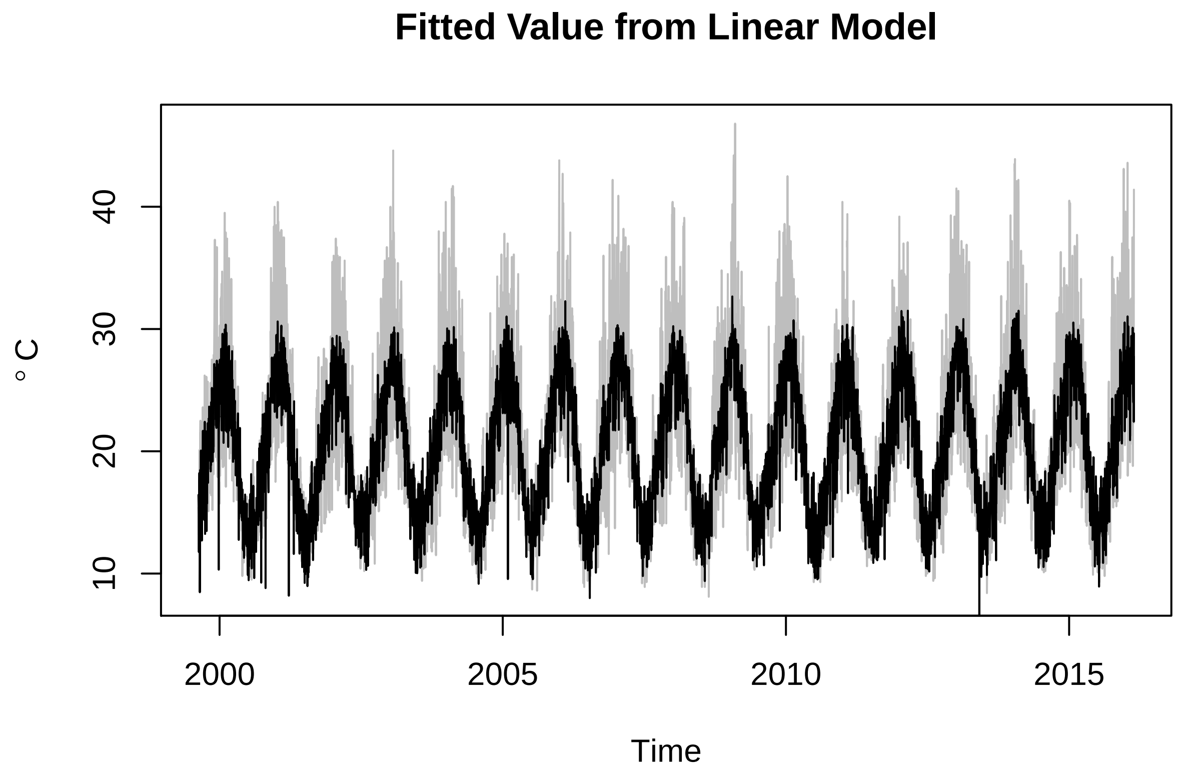
```r
# Fit a linear trend with seasonality using tslm()
tslm_max = tslm(ts_max ~ trend + ts_rainfall + ts_sunshine + ts_wind + fourier(ts_max,
    K = 2))
summary(tslm_max)
```

```
##
## Call:
## tslm(formula = ts_max ~ trend + ts_rainfall + ts_sunshine + ts_wind +
##     fourier(ts_max, K = 2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2130  -2.6097  -0.2136   2.1744  16.7789
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.399e+01  2.082e-01  67.215  < 2e-16 ***
## trend                         1.051e-04  2.913e-05   3.608 0.000311 ***
## ts_rainfall                   -1.428e-01  1.103e-02 -12.956  < 2e-16 ***
## ts_sunshine                    4.364e-01  1.461e-02  29.870  < 2e-16 ***
## ts_wind                        7.269e-02  3.192e-03  22.774  < 2e-16 ***
## fourier(ts_max, K = 2)S1-365  2.338e+00  7.663e-02  30.508  < 2e-16 ***
## fourier(ts_max, K = 2)C1-365 -5.446e+00  7.522e-02 -72.401  < 2e-16 ***
## fourier(ts_max, K = 2)S2-365  4.179e-01  7.187e-02   5.815 6.39e-09 ***
## fourier(ts_max, K = 2)C2-365  3.168e-01  7.210e-02   4.394 1.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.94 on 6025 degrees of freedom
## Multiple R-squared:  0.6409, Adjusted R-squared:  0.6404
## F-statistic:  1344 on 8 and 6025 DF,  p-value: < 2.2e-16
```

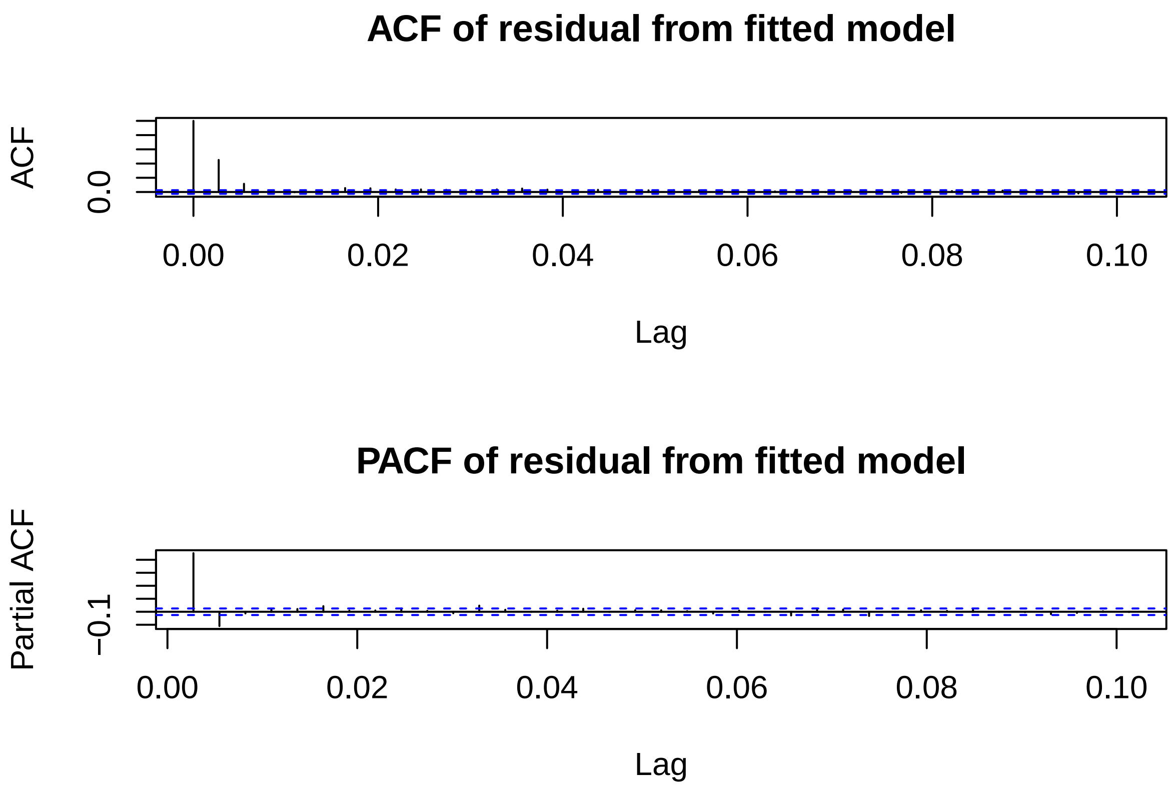
```r
AIC(tslm_max)  # AIC = 33683.01
```

```
## [1] 33683.01
```

```r
plot(ts_max, col = "grey", main = "Fitted Value from Linear Model", ylab =
↪  expression(degree ~
    C))
lines(tslm_max$fitted.values)
```



```r
## Model diagnosis
par(mfrow = c(2, 1))
acf(tslm_max$residuals, main = "ACF of residual from fitted model")  # lag q = 2
pacf(tslm_max$residuals, main = "PACF of residual from fitted model")  # lag p = 2
```



```r
Box.test(tslm_max$residuals, type = "Lj")  #autocorrelation different from 0
```

```
##
##  Box-Ljung test
##
## data:  tslm_max$residuals
## X-squared = 1228.3, df = 1, p-value < 2.2e-16
```

```r
dwtest(tslm_max, alternative = "two")
```

```
##
##  Durbin-Watson test
##
## data:  tslm_max
## DW = 1.0964, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is not 0
```

```
bgtest(tslm_max, 20)
```

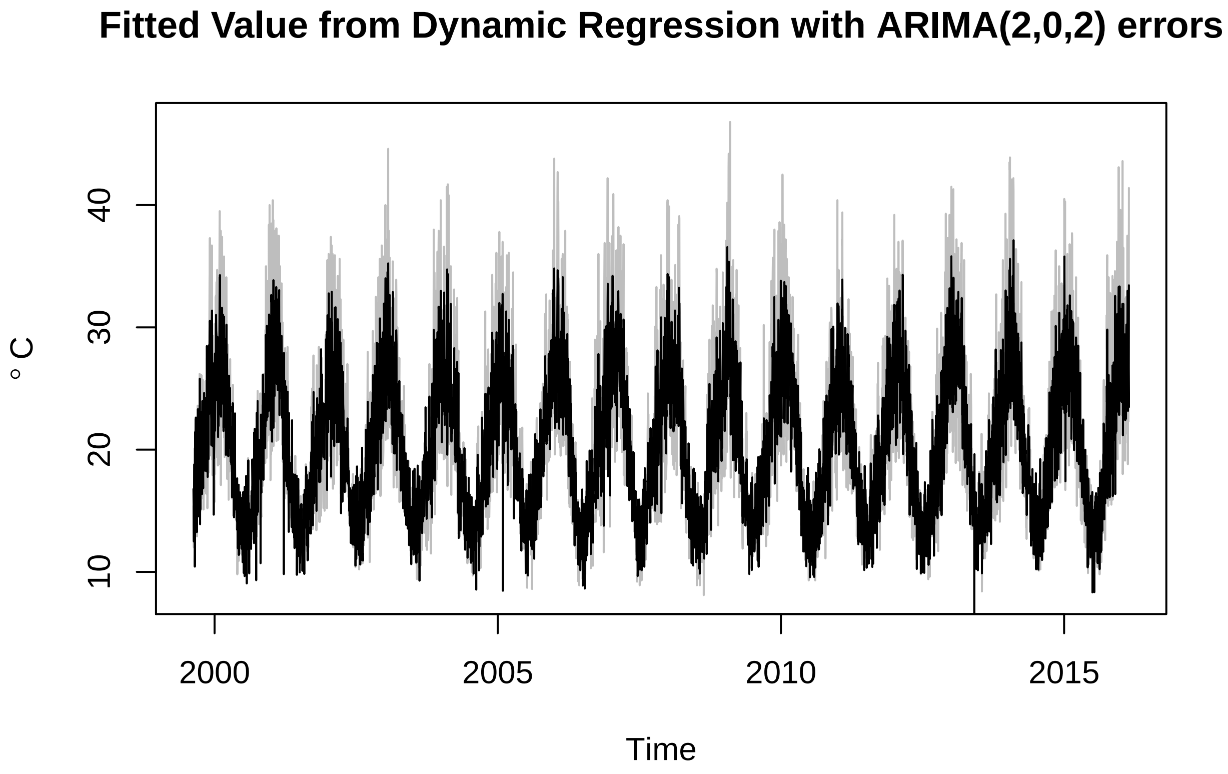
```
##
##  Breusch-Godfrey test for serial correlation of order up to 20
##
## data:  tslm_max
## LM test = 1365.2, df = 20, p-value < 2.2e-16
```

2. Dynamic Regression with ARIMA(2,0,2) error

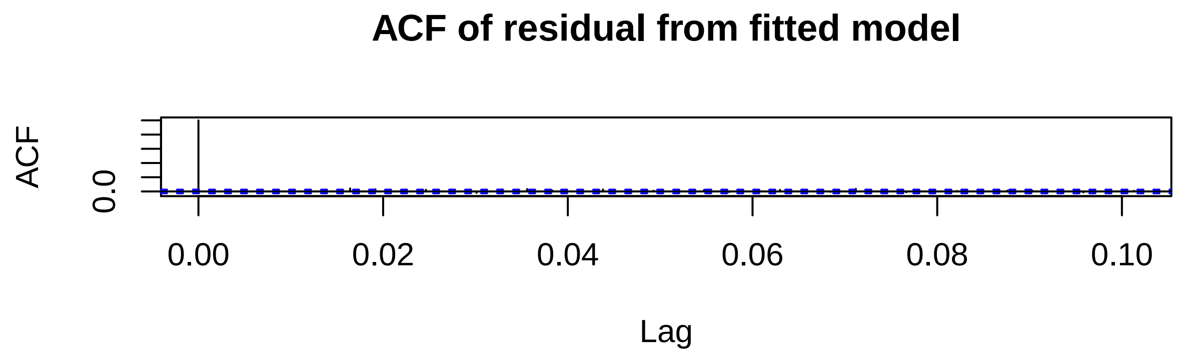
```
# Fit a dynamic regression to capture the dynamics left in the residuals
dr_max = Arima(ts_max, xreg = cbind(ts_rainfall, ts_sunshine, ts_wind, fourier(ts_max,
    K = 2)), order = c(2, 0, 2))
summary(dr_max)  # AIC = 32213.94
```

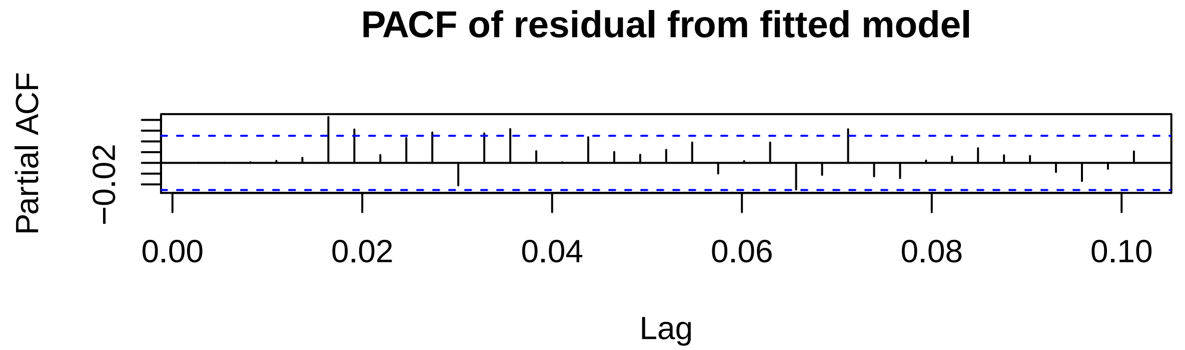
```
## Series: ts_max
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##          ar1      ar2     ma1     ma2  intercept  ts_rainfall  ts_sunshine
##       0.3119  -0.0777  0.2000  0.0720    14.4956      -0.1114       0.3825
## s.e.  0.2973   0.0948  0.2974  0.0658     0.1857       0.0094       0.0129
##       ts_wind  fourier(ts_max, K = 2).S1-365  fourier(ts_max, K = 2).C1-365
##        0.0754                         2.4065                        -5.5318
## s.e.   0.0030                         0.1081                         0.1071
##       fourier(ts_max, K = 2).S2-365  fourier(ts_max, K = 2).C2-365
##                              0.4200                         0.3435
## s.e.                         0.1055                         0.1055
##
## sigma^2 estimated as 12.16:  log likelihood=-16093.97
## AIC=32213.94   AICc=32214   BIC=32301.11
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.001506474 3.484254 2.624239 -2.377646 12.90383 0.5519211
##                      ACF1
## Training set 0.0001884346
```

```
plot(ts_max, col = "grey", main = "Fitted Value from Dynamic Regression with ARIMA(2,0,2)
↪  errors",
    ylab = expression(degree ~ C))
lines(dr_max$fitted)
```



```r
par(mfrow = c(2, 1))
acf(dr_max$residuals, main = "ACF of residual from fitted model")
pacf(dr_max$residuals, main = "PACF of residual from fitted model")
```





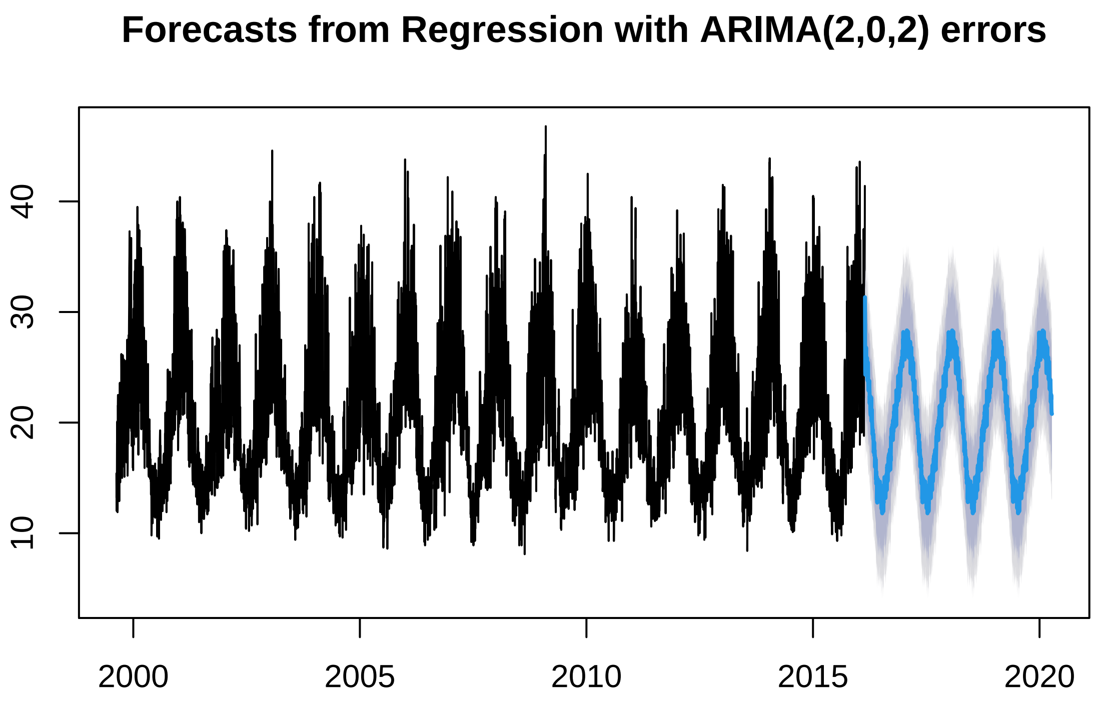
```
Box.test(dr_max$residuals)
```

```
##
##  Box-Pierce test
##
## data:  dr_max$residuals
## X-squared = 0.00021425, df = 1, p-value = 0.9883
```

```
## Construct a function to calculate p-value for fitted models
p_value = function(model) {
    t_fit = model$coef/(sqrt(diag(model$var.coef)))
    p_fit = 2 * pnorm(abs(t_fit), mean = 0, sd = 1, lower.tail = FALSE)

    return(p_fit)
}
p_value(dr_max)
```

```
##                           ar1                           ar2
##                  2.941101e-01                  4.124677e-01
##                           ma1                           ma2
##                  5.012829e-01                  2.741662e-01
##                     intercept                   ts_rainfall
##                  0.000000e+00                  2.011271e-32
##                   ts_sunshine                       ts_wind
##                  1.178194e-193                 2.610144e-141
## fourier(ts_max, K = 2).S1-365 fourier(ts_max, K = 2).C1-365
##                  1.079688e-109                  0.000000e+00
## fourier(ts_max, K = 2).S2-365 fourier(ts_max, K = 2).C2-365
##                   6.823194e-05                  1.131803e-03
```

```
# Forecast using fitted model
fcast_rainfall = forecast(ts_rainfall, method = "ets", h = 1508)
fcast_sunshine = forecast(ts_sunshine, method = "ets", h = 1508)
fcast_wind = forecast(ts_wind, method = "ets", h = 1508)

fcast_xreg = cbind(fcast_rainfall$mean, fcast_sunshine$mean, fcast_wind$mean,
↪   fourier(ts_max,
    K = 2, h = 1508))
colnames(fcast_xreg) = names(dr_max$coef)[-c(1:5)]

fcast1 = forecast(dr_max, xreg = fcast_xreg, h = 1508)
par(mfrow = c(1, 1))
plot(fcast1)
```



```
accuracy(fcast1, ts_test[, "max"])
```

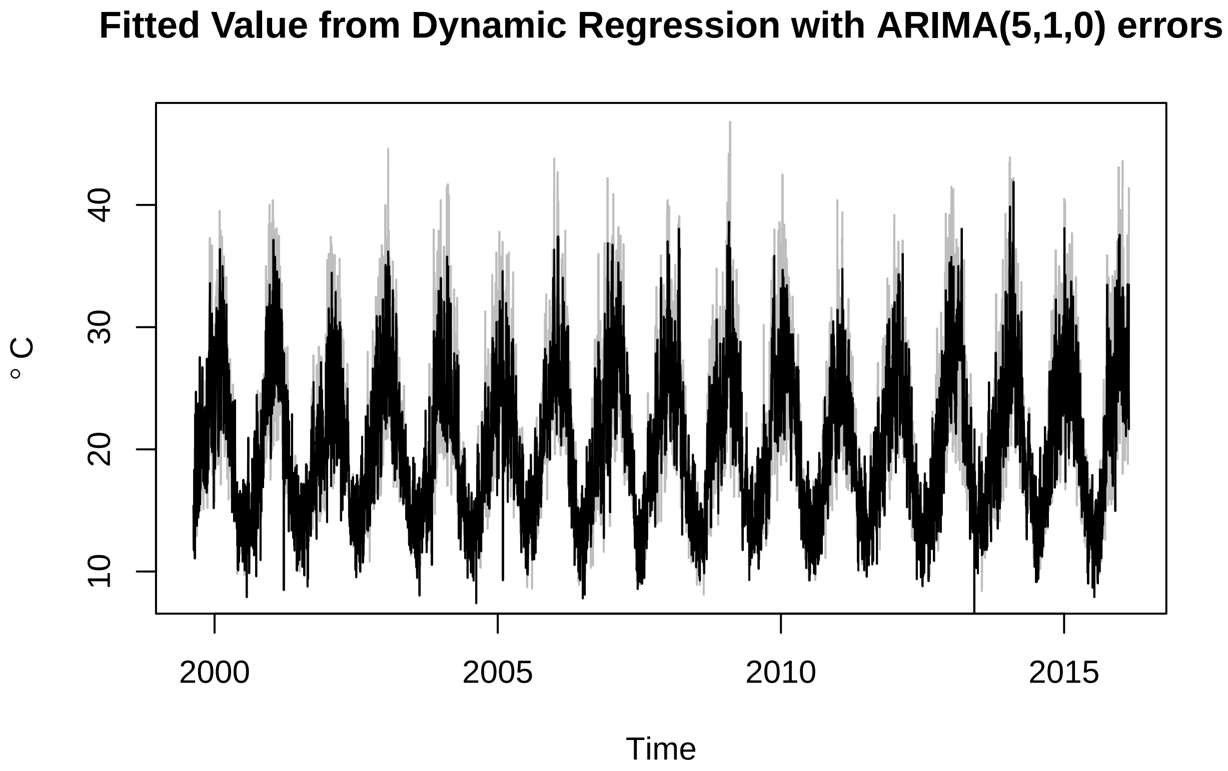
```
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.001506474 3.484254 2.624239 -2.3776465 12.90383 0.5519211
## Test set      0.630405671 4.546492 3.455661 -0.8162259 16.01995 0.7267829
##                      ACF1 Theil's U
## Training set 0.0001884346        NA
## Test set     0.4054704347 0.9815607
```

3. Dynamic Regression with ARIMA(5,1,0) error

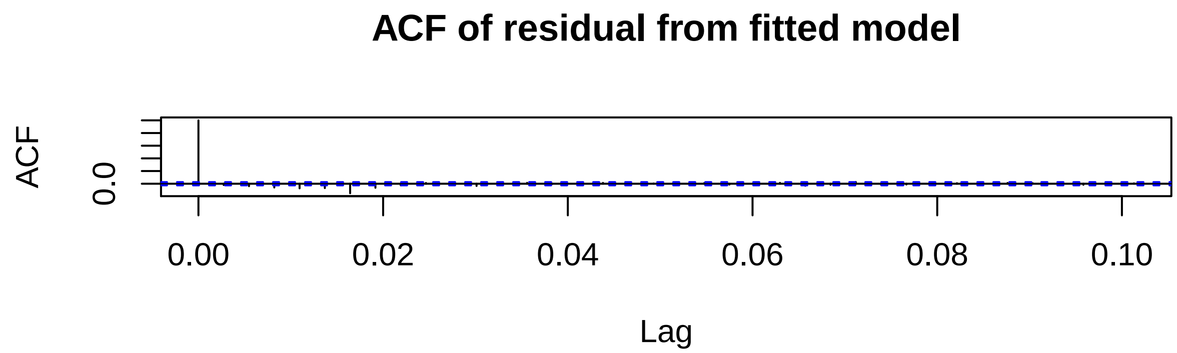
```
# Fit another dynamic regression using auto arima
dr2_max = auto.arima(ts_max, xreg = cbind(ts_rainfall, ts_sunshine, ts_wind,
↪  fourier(ts_max,
    K = 2)))

summary(dr2_max)  # ARIMA(5,1,0), AIC = 32894.15
```

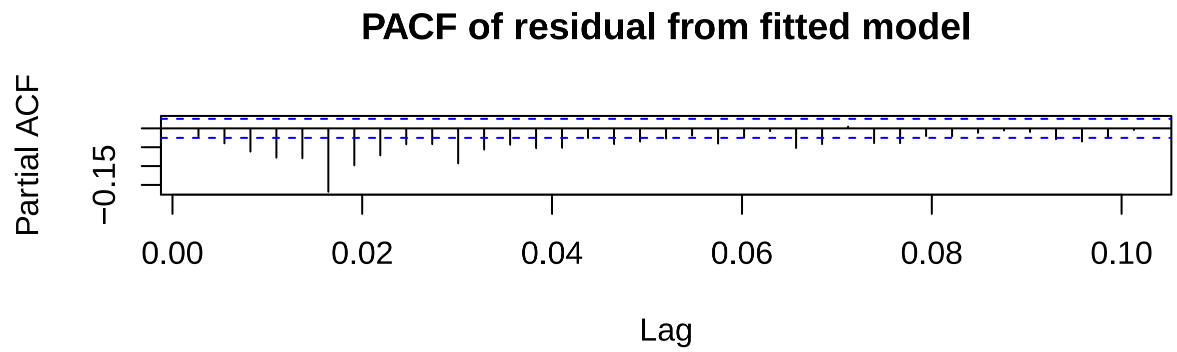
```
## Series: ts_max
## Regression with ARIMA(5,1,0) errors
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5  ts_rainfall  ts_sunshine
##       -0.3593  -0.4016  -0.3344  -0.2413  -0.1734      -0.1073       0.3773
## s.e.   0.0128   0.0132   0.0135   0.0132   0.0127       0.0093       0.0127
##       ts_wind  fourier(ts_max, K = 2).S1-365  fourier(ts_max, K = 2).C1-365
##        0.0768                         2.3077                        -5.5230
## s.e.   0.0030                         1.5564                         1.5568
##       fourier(ts_max, K = 2).S2-365  fourier(ts_max, K = 2).C2-365
##                              0.4726                         0.3174
## s.e.                         0.7789                         0.7795
##
## sigma^2 estimated as 13.63:  log likelihood=-16434.07
## AIC=32894.15   AICc=32894.21   BIC=32981.31
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 0.002838698 3.687538 2.751103 -2.09439 13.55714 0.5786026
##                     ACF1
## Training set -0.02089691
```

```
plot(ts_max, col = "grey", main = "Fitted Value from Dynamic Regression with ARIMA(5,1,0)
↪  errors",
    ylab = expression(degree ~ C))
lines(dr2_max$fitted)
```



```
par(mfrow = c(2, 1))
acf(dr2_max$residuals, main = "ACF of residual from fitted model")
pacf(dr2_max$residuals, main = "PACF of residual from fitted model")
```

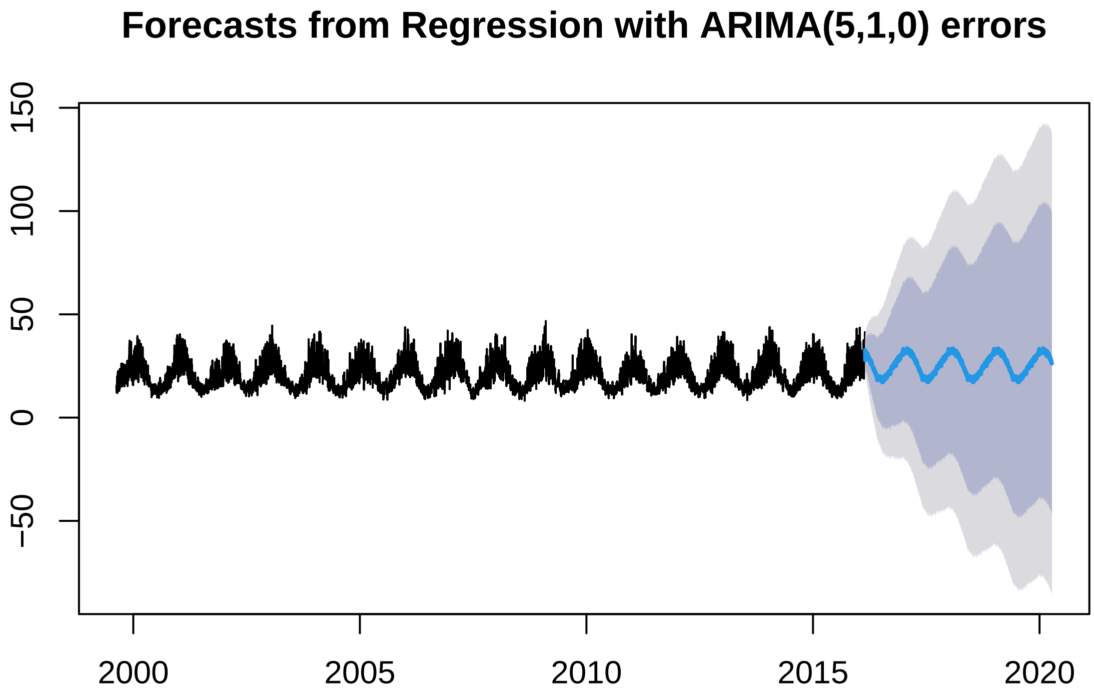

ACF of residual from fitted model


PACF of residual from fitted model

```
Box.test(dr2_max$residuals)
```

```
##
##  Box-Pierce test
##
## data:  dr2_max$residuals
## X-squared = 2.6349, df = 1, p-value = 0.1045
```

```
# Forecast using fitted model
fcast2 = forecast(dr2_max, xreg = fcast_xreg, h = 1508)
par(mfrow = c(1, 1))
plot(fcast2)
```



```r
accuracy(fcast2, ts_test[, "max"])
```

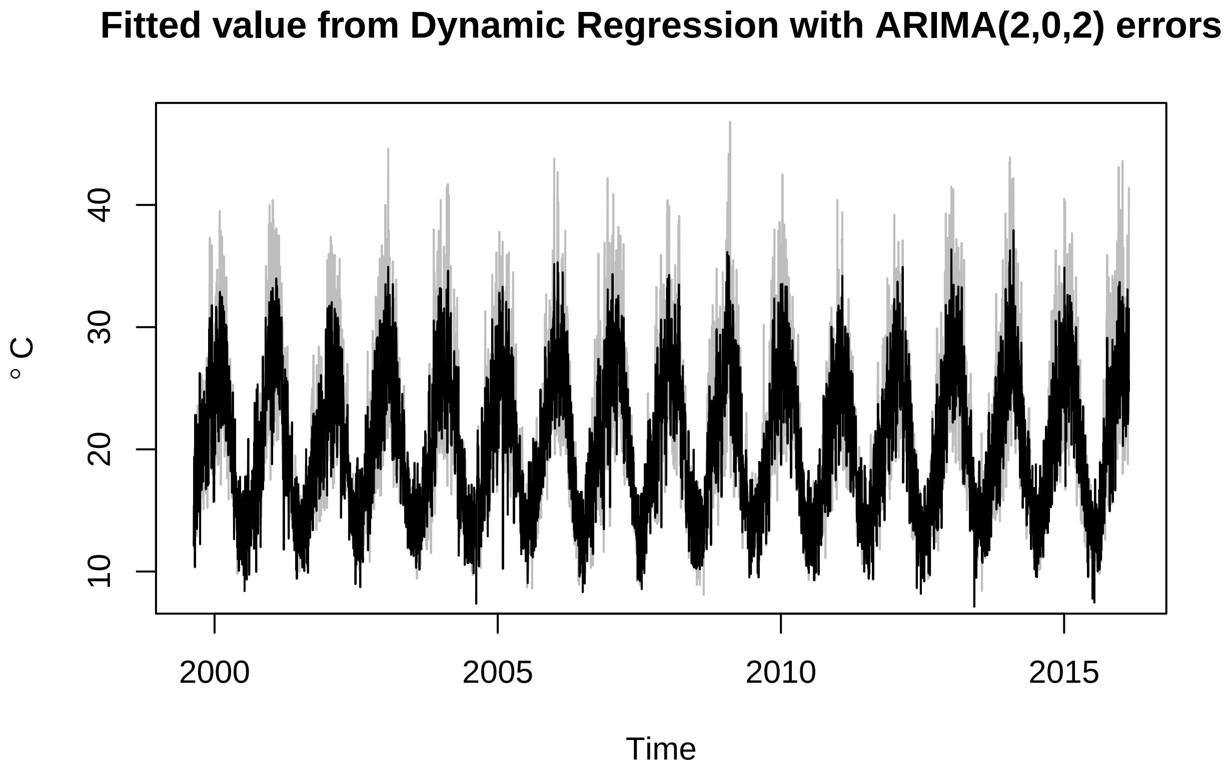
```
##                        ME     RMSE      MAE       MPE     MAPE      MASE
## Training set  0.002838698 3.687538 2.751103  -2.09439 13.55714 0.5786026
## Test set     -4.654307073 6.478497 5.638558 -28.67872 31.51016 1.1858823
##                     ACF1 Theil's U
## Training set -0.02089691        NA
## Test set      0.40549696  1.760783
```

4. Dynamic Regression with ARIMA(2,0,2) error and lag predictors

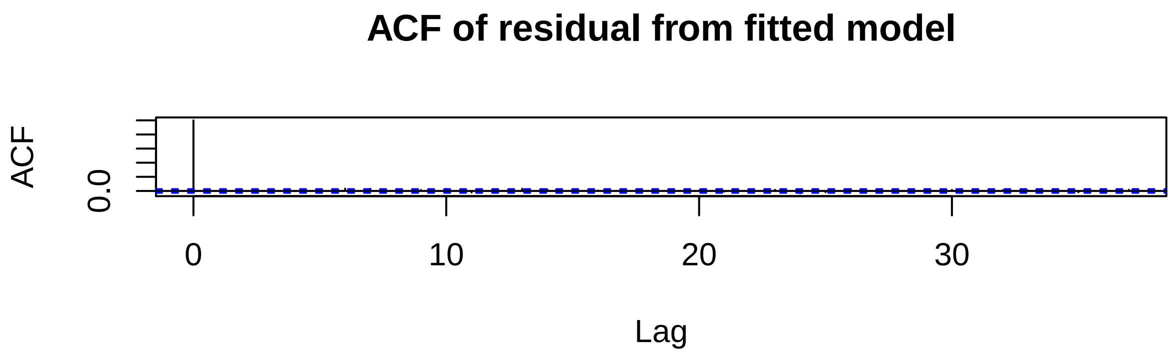
```r
# Fit a dynamic regression with lag values
len_train = length(ts_train[, "max"])
lag1_sunshine = c(NA, ts_sunshine[1:(len_train - 1)])
lag2_sunshine = c(rep(NA, 2), ts_sunshine[1:(len_train - 2)])
lag1_rainfall = c(NA, ts_rainfall[1:(len_train - 1)])
lag2_rainfall = c(rep(NA, 2), ts_rainfall[1:(len_train - 2)])
lag1_wind = c(NA, ts_wind[1:(len_train - 1)])
lag2_wind = c(rep(NA, 2), ts_wind[1:(len_train - 2)])

drlag_max = Arima(ts_max, xreg = cbind(ts_rainfall, lag1_rainfall, lag2_rainfall,
    ts_sunshine, lag1_sunshine, lag2_sunshine, ts_wind, lag1_wind, lag2_wind,
↪   fourier(ts_max,
        K = 2)), order = c(2, 0, 2))
summary(drlag_max)  #AIC = 31560.4
```

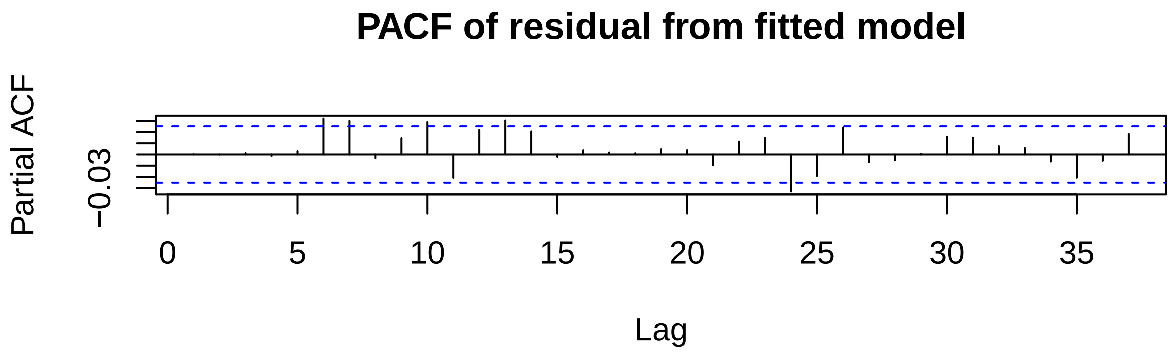
```
## Series: ts_max
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##          ar1      ar2     ma1     ma2  intercept  ts_rainfall  lag1_rainfall
##       0.4347  -0.0893  0.0498  0.0541    15.4300      -0.0644        -0.0229
## s.e.  0.2797   0.1024  0.2797  0.0464     0.3204       0.0097         0.0102
##       lag2_rainfall  ts_sunshine  lag1_sunshine  lag2_sunshine  ts_wind
##             -0.0258       0.4175         0.2432         0.1182   0.0595
## s.e.         0.0094       0.0127         0.0136         0.0132   0.0029
##       lag1_wind  lag2_wind  fourier(ts_max, K = 2).S1-365
##         -0.0502    -0.0083                         2.0346
## s.e.     0.0029     0.0029                         0.1118
##       fourier(ts_max, K = 2).C1-365  fourier(ts_max, K = 2).S2-365
##                             -4.7925                         0.3369
## s.e.                         0.1099                         0.1015
##       fourier(ts_max, K = 2).C2-365
##                              0.3365
## s.e.                         0.1021
##
## sigma^2 estimated as 10.92:  log likelihood=-15761.2
## AIC=31560.4   AICc=31560.52   BIC=31687.79
##
## Training set error measures:
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.0001517959 3.300164 2.508675 -2.084281 12.45151 0.5276159
##                      ACF1
## Training set 0.0001495279
```

```
plot(ts_max, col = "grey", main = "Fitted value from Dynamic Regression with ARIMA(2,0,2)
↪   errors",
    ylab = expression(degree ~ C))
lines(drlag_max$fitted)
```



```r
par(mfrow = c(2, 1))
acf(drlag_max$residuals[3:len_train], main = "ACF of residual from fitted model")
pacf(drlag_max$residuals[3:len_train], main = "PACF of residual from fitted model")
```

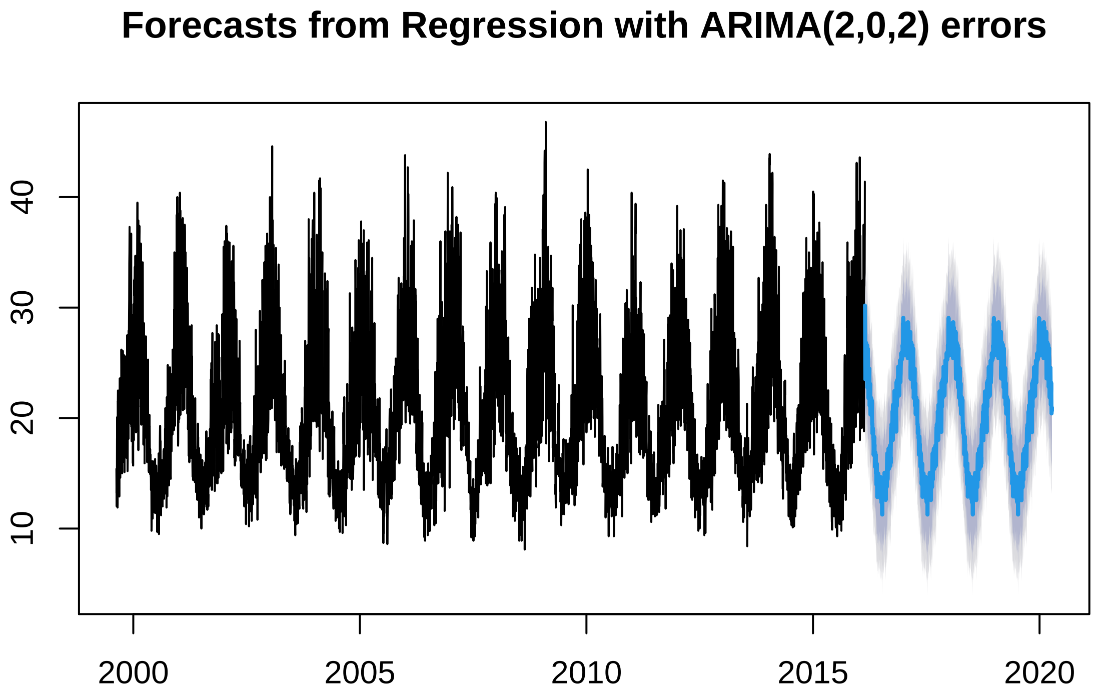

ACF of residual from fitted model


PACF of residual from fitted model

```
Box.test(drlag_max$residuals)
```

```
##
##  Box-Pierce test
##
## data:  drlag_max$residuals
## X-squared = 0.00013487, df = 1, p-value = 0.9907
```

```
# Forecast using fitted model
fcast_lag1_rainfall = c(ts_sunshine[6034], fcast_rainfall$mean[1:1507])
fcast_lag2_rainfall = c(ts_sunshine[6033:6034], fcast_rainfall$mean[1:1506])
fcast_lag1_sunshine = c(ts_sunshine[6034], fcast_sunshine$mean[1:1507])
fcast_lag2_sunshine = c(ts_sunshine[6033:6034], fcast_sunshine$mean[1:1506])
fcast_lag1_wind = c(ts_wind[6034], fcast_wind$mean[1:1507])
fcast_lag2_wind = c(ts_wind[6033:6034], fcast_wind$mean[1:1506])

fcast_lag_xreg = cbind(fcast_rainfall$mean, fcast_lag1_rainfall, fcast_lag2_rainfall,
    fcast_sunshine$mean, fcast_lag1_sunshine, fcast_lag2_sunshine, fcast_wind$mean,
    fcast_lag1_wind, fcast_lag2_wind, fourier(ts_max, K = 2, h = 1508))
colnames(fcast_lag_xreg) = names(drlag_max$coef)[-c(1:5)]

fcast3 = forecast(drlag_max, xreg = fcast_lag_xreg, h = 1508)
par(mfrow = c(1, 1))
plot(fcast3)
```



```r
accuracy(fcast3, ts_test[, "max"])
```

```
##                          ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.0001517959 3.300164 2.508675 -2.084281 12.45151 0.5276159
## Test set      0.5938947740 4.569709 3.483299 -0.980496 16.21407 0.7325955
##                      ACF1 Theil's U
## Training set 0.0001495279        NA
## Test set     0.4080117323 0.9925416
```

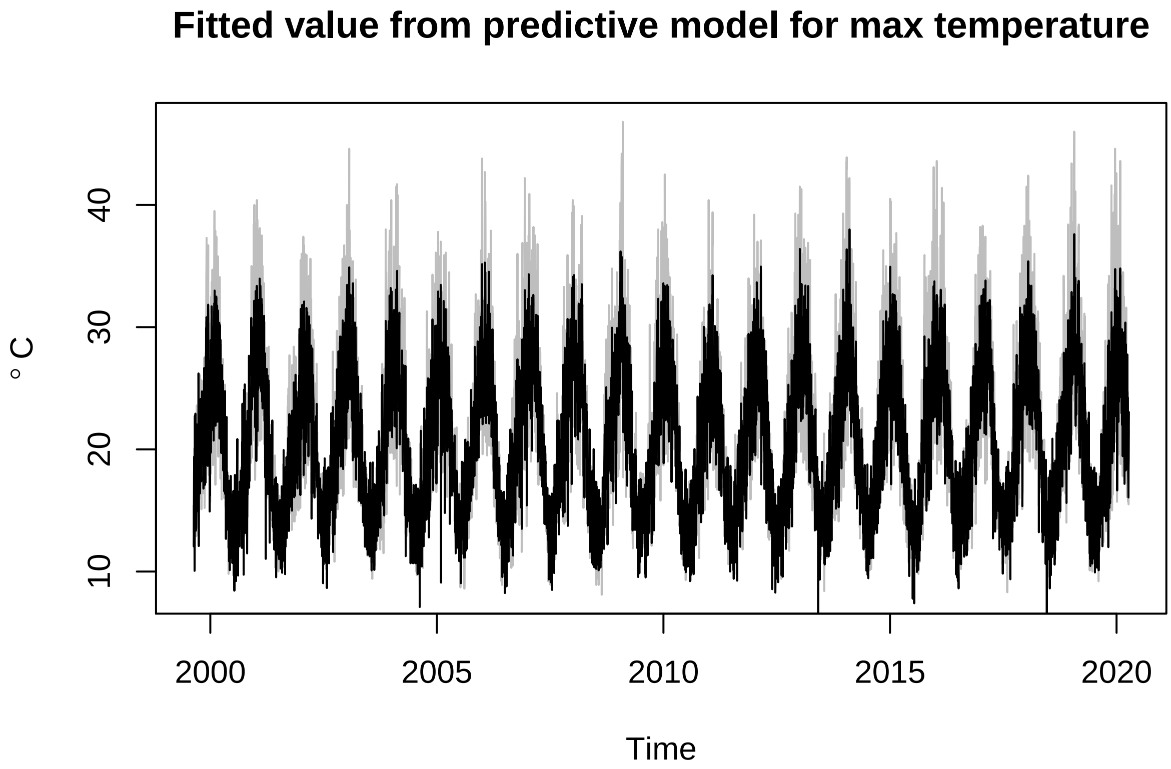
Use the best model (No.4) to build full temperature forecast

```r
full_len = length(ts_full[, "max"])
full_max = ts_full[, "max"]
full_sunshine = ts_full[, "sunshine"]
full_rainfall = ts_full[, "rainfall"]
full_wind = ts_full[, "wind"]

full_lag1_sunshine = c(NA, full_sunshine[1:(full_len - 1)])
full_lag2_sunshine = c(rep(NA, 2), full_sunshine[1:(full_len - 2)])
full_lag1_rainfall = c(NA, full_rainfall[1:(full_len - 1)])
full_lag2_rainfall = c(rep(NA, 2), full_rainfall[1:(full_len - 2)])
full_lag1_wind = c(NA, full_wind[1:(full_len - 1)])
full_lag2_wind = c(rep(NA, 2), full_wind[1:(full_len - 2)])
```

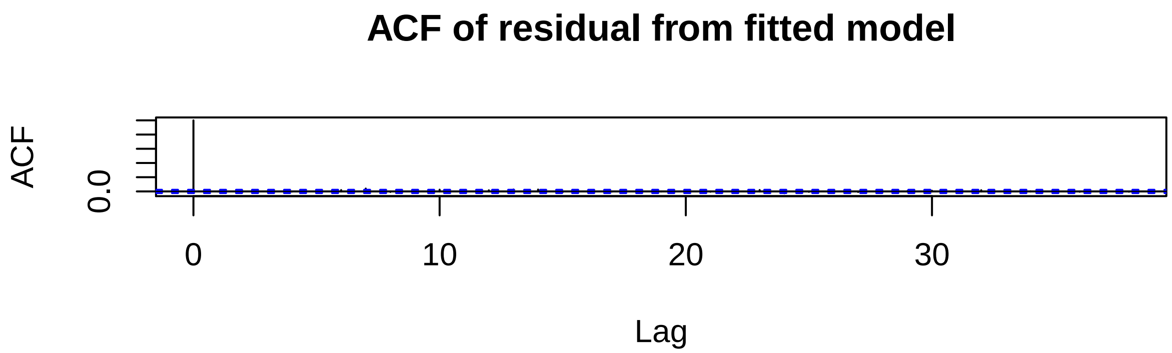
```r
finaldr_max = Arima(full_max, xreg = cbind(full_sunshine, full_lag1_sunshine,
↪  full_lag2_sunshine,
    full_rainfall, full_lag1_rainfall, full_lag2_rainfall, full_wind, full_lag1_wind,
    full_lag2_wind, fourier(full_max, K = 2)), order = c(2, 0, 2))
summary(finaldr_max)
```

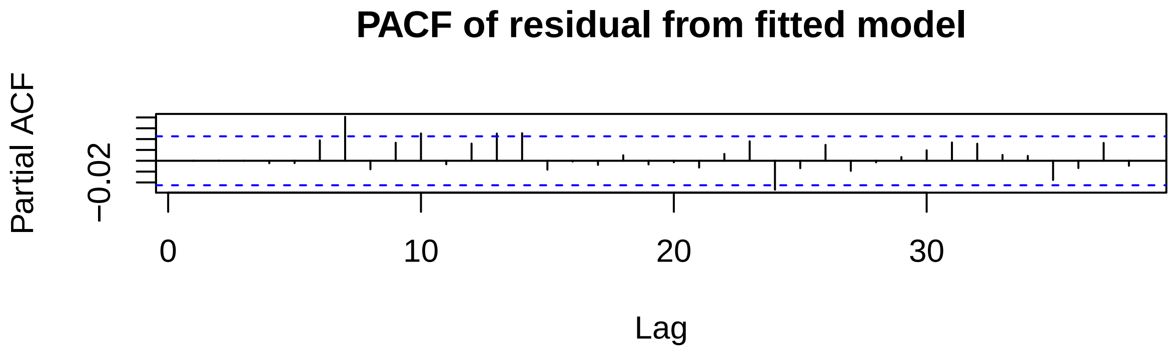
```
## Series: full_max
## Regression with ARIMA(2,0,2) errors
##
## Coefficients:
##          ar1     ar2     ma1     ma2  intercept  full_sunshine
##       0.2189  0.0077  0.2545  0.0523    15.4908         0.4090
## s.e.  1.1741  0.4071  1.1740  0.1503     0.2885         0.0114
##       full_lag1_sunshine  full_lag2_sunshine  full_rainfall  full_lag1_rainfall
##                   0.2382              0.1203        -0.0749             -0.0305
## s.e.              0.0122              0.0119         0.0087              0.0091
##       full_lag2_rainfall  full_wind  full_lag1_wind  full_lag2_wind
##                  -0.0253     0.0620         -0.0495         -0.0091
## s.e.              0.0084     0.0026          0.0026          0.0026
##       fourier(full_max, K = 2).S1-365  fourier(full_max, K = 2).C1-365
##                                2.0328                          -4.9460
## s.e.                           0.1004                           0.0989
##       fourier(full_max, K = 2).S2-365  fourier(full_max, K = 2).C2-365
##                                0.3476                           0.3008
## s.e.                           0.0914                           0.0923
##
## sigma^2 estimated as 11.07:  log likelihood=-19754.23
## AIC=39546.45   AICc=39546.55   BIC=39678.08
##
## Training set error measures:
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.0001978426 3.323339 2.533286 -2.096745 12.53738 0.5317751
##                      ACF1
## Training set 1.017968e-05
```

```r
plot(full_max, col = "grey", main = "Fitted value from predictive model for max
↪  temperature",
    ylab = expression(degree ~ C))
lines(finaldr_max$fitted)
```



```r
par(mfrow = c(2, 1))
acf(finaldr_max$residuals[3:full_len], main = "ACF of residual from fitted model")
pacf(finaldr_max$residuals[3:full_len], main = "PACF of residual from fitted model")
```


ACF of residual from fitted model

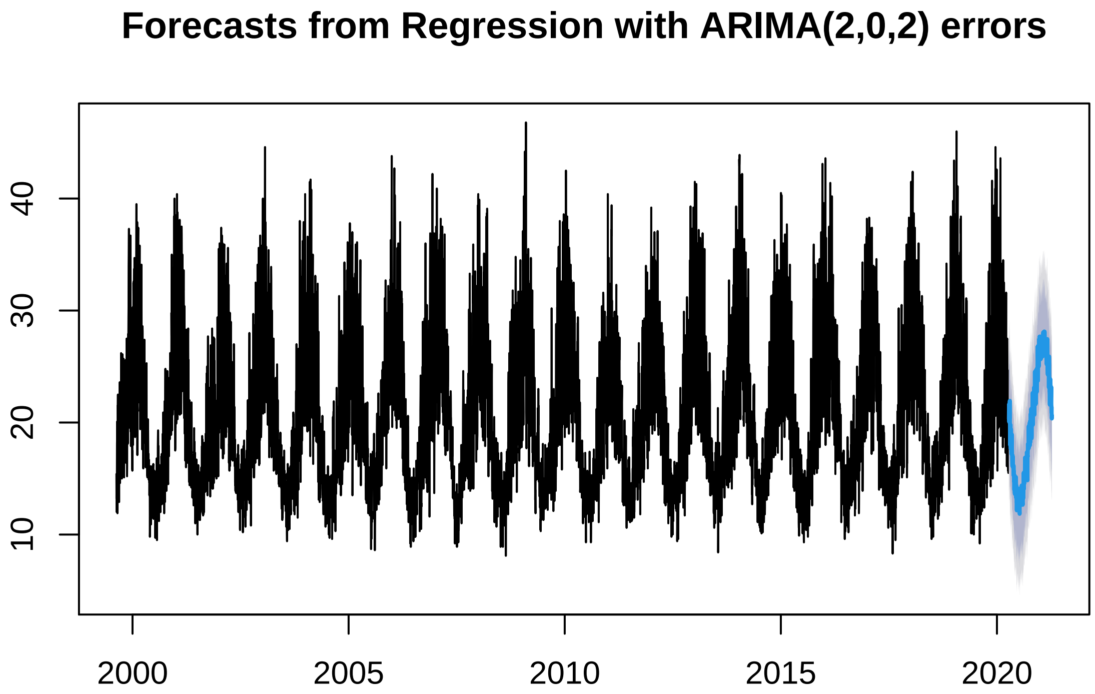

PACF of residual from fitted model

```
Box.test(finaldr_max$residuals)
```

```
##
##  Box-Pierce test
##
## data:  finaldr_max$residuals
## X-squared = 7.8134e-07, df = 1, p-value = 0.9993
```

```
## build forecast
fcast_full_sunshine = forecast(full_sunshine, method = "ets", h = 365)
fcast_full_rainfall = forecast(full_rainfall, method = "ets", h = 365)
fcast_full_wind = forecast(full_wind, method = "ets", h = 365)
fcast_full_lag1_sunshine = c(full_sunshine[7542], fcast_full_sunshine$mean[1:364])
fcast_full_lag2_sunshine = c(full_sunshine[7541:7542], fcast_full_sunshine$mean[1:363])
fcast_full_lag1_rainfall = c(full_rainfall[7542], fcast_full_rainfall$mean[1:364])
fcast_full_lag2_rainfall = c(full_rainfall[7541:7542], fcast_full_rainfall$mean[1:363])
fcast_full_lag1_wind = c(full_wind[7542], fcast_full_wind$mean[1:364])
fcast_full_lag2_wind = c(full_wind[7541:7542], fcast_full_wind$mean[1:363])

fcast_full_xreg = cbind(fcast_full_sunshine$mean, fcast_full_lag1_sunshine,
↪  fcast_full_lag2_sunshine,
    fcast_full_rainfall$mean, fcast_full_lag1_rainfall, fcast_full_lag2_rainfall,
    fcast_full_wind$mean, fcast_full_lag1_wind, fcast_full_lag2_wind, fourier(full_max,
        K = 2, h = 365))
colnames(fcast_full_xreg) = names(finaldr_max$coef)[-c(1:5)]
```

```r
fcast_full_max = forecast(finaldr_max, xreg = fcast_full_xreg, h = 365)
par(mfrow = c(1, 1))
plot(fcast_full_max)
```



```r
max_temperature_result = data.frame(seq(as.Date("2020-04-11"), by = "days", length.out =
↪   365),
    round(fcast_full_max$mean, 2))
colnames(max_temperature_result) = c("Date", "Forecasted Max Temperature")
```

```r
kable(max_temperature_result)
```

Date	Forecasted Max Temperature
2020-04-11	20.33
2020-04-12	21.22
2020-04-13	21.65
2020-04-14	20.05
2020-04-15	20.47
2020-04-16	20.50
2020-04-17	20.35
2020-04-18	21.30

Date	Forecasted Max Temperature
2020-04-19	21.94
2020-04-20	19.57
2020-04-21	18.00
2020-04-22	18.88
2020-04-23	19.09
2020-04-24	18.97
2020-04-25	19.80
2020-04-26	18.57
2020-04-27	17.66
2020-04-28	17.97
2020-04-29	19.40
2020-04-30	18.79
2020-05-01	18.05
2020-05-02	18.22
2020-05-03	17.20
2020-05-04	18.19
2020-05-05	18.26
2020-05-06	18.94
2020-05-07	17.82
2020-05-08	17.30
2020-05-09	16.55
2020-05-10	16.24
2020-05-11	16.10
2020-05-12	16.97
2020-05-13	17.05
2020-05-14	16.72
2020-05-15	16.88
2020-05-16	16.20
2020-05-17	15.97
2020-05-18	16.00
2020-05-19	16.12
2020-05-20	15.30
2020-05-21	16.15
2020-05-22	15.57
2020-05-23	15.64
2020-05-24	14.92
2020-05-25	14.76
2020-05-26	15.04
2020-05-27	13.98
2020-05-28	15.00
2020-05-29	15.30
2020-05-30	14.43
2020-05-31	13.66
2020-06-01	14.15
2020-06-02	14.08
2020-06-03	13.59
2020-06-04	13.63
2020-06-05	14.47
2020-06-06	14.27
2020-06-07	14.32
2020-06-08	14.11
2020-06-09	15.11

Date	Forecasted Max Temperature
2020-06-10	15.13
2020-06-11	14.36
2020-06-12	13.26
2020-06-13	14.39
2020-06-14	13.57
2020-06-15	12.88
2020-06-16	12.77
2020-06-17	12.16
2020-06-18	12.66
2020-06-19	13.24
2020-06-20	13.85
2020-06-21	13.53
2020-06-22	14.04
2020-06-23	12.88
2020-06-24	13.28
2020-06-25	13.91
2020-06-26	13.41
2020-06-27	12.77
2020-06-28	13.45
2020-06-29	12.19
2020-06-30	13.05
2020-07-01	14.11
2020-07-02	13.55
2020-07-03	12.88
2020-07-04	12.15
2020-07-05	13.09
2020-07-06	13.47
2020-07-07	12.66
2020-07-08	12.55
2020-07-09	13.38
2020-07-10	12.08
2020-07-11	11.85
2020-07-12	13.27
2020-07-13	12.54
2020-07-14	13.18
2020-07-15	13.76
2020-07-16	13.82
2020-07-17	12.98
2020-07-18	12.76
2020-07-19	13.97
2020-07-20	13.28
2020-07-21	14.01
2020-07-22	13.18
2020-07-23	12.80
2020-07-24	13.65
2020-07-25	13.25
2020-07-26	13.35
2020-07-27	13.02
2020-07-28	14.38
2020-07-29	13.80
2020-07-30	14.21
2020-07-31	13.60

Date	Forecasted Max Temperature
2020-08-01	14.24
2020-08-02	12.98
2020-08-03	13.21
2020-08-04	13.99
2020-08-05	12.88
2020-08-06	12.65
2020-08-07	13.29
2020-08-08	13.85
2020-08-09	14.28
2020-08-10	13.22
2020-08-11	13.95
2020-08-12	13.82
2020-08-13	14.88
2020-08-14	15.51
2020-08-15	14.67
2020-08-16	14.06
2020-08-17	14.76
2020-08-18	14.59
2020-08-19	14.53
2020-08-20	14.61
2020-08-21	15.21
2020-08-22	15.56
2020-08-23	15.13
2020-08-24	15.11
2020-08-25	16.23
2020-08-26	16.87
2020-08-27	15.17
2020-08-28	15.47
2020-08-29	15.93
2020-08-30	16.61
2020-08-31	15.55
2020-09-01	15.01
2020-09-02	15.94
2020-09-03	16.29
2020-09-04	16.57
2020-09-05	16.69
2020-09-06	16.60
2020-09-07	16.73
2020-09-08	15.71
2020-09-09	15.25
2020-09-10	17.35
2020-09-11	16.64
2020-09-12	16.20
2020-09-13	16.94
2020-09-14	15.73
2020-09-15	14.84
2020-09-16	17.05
2020-09-17	16.95
2020-09-18	17.47
2020-09-19	18.09
2020-09-20	17.73
2020-09-21	18.07

Date	Forecasted Max Temperature
2020-09-22	17.92
2020-09-23	18.15
2020-09-24	17.67
2020-09-25	18.98
2020-09-26	19.69
2020-09-27	17.74
2020-09-28	17.85
2020-09-29	17.90
2020-09-30	18.72
2020-10-01	19.36
2020-10-02	19.95
2020-10-03	19.13
2020-10-04	18.77
2020-10-05	18.48
2020-10-06	17.97
2020-10-07	20.13
2020-10-08	18.70
2020-10-09	18.22
2020-10-10	19.01
2020-10-11	19.47
2020-10-12	20.21
2020-10-13	20.79
2020-10-14	19.97
2020-10-15	19.87
2020-10-16	19.96
2020-10-17	21.22
2020-10-18	20.33
2020-10-19	20.04
2020-10-20	18.58
2020-10-21	18.82
2020-10-22	19.83
2020-10-23	20.66
2020-10-24	20.64
2020-10-25	20.21
2020-10-26	20.48
2020-10-27	21.17
2020-10-28	22.63
2020-10-29	22.62
2020-10-30	21.42
2020-10-31	19.80
2020-11-01	19.96
2020-11-02	21.45
2020-11-03	21.89
2020-11-04	21.41
2020-11-05	21.43
2020-11-06	21.26
2020-11-07	21.75
2020-11-08	21.22
2020-11-09	21.66
2020-11-10	23.13
2020-11-11	22.90
2020-11-12	21.50

Date	Forecasted Max Temperature
2020-11-13	20.70
2020-11-14	21.53
2020-11-15	22.17
2020-11-16	23.28
2020-11-17	24.51
2020-11-18	23.34
2020-11-19	24.56
2020-11-20	23.74
2020-11-21	22.37
2020-11-22	23.04
2020-11-23	21.93
2020-11-24	21.64
2020-11-25	22.01
2020-11-26	22.03
2020-11-27	23.83
2020-11-28	24.71
2020-11-29	24.70
2020-11-30	24.12
2020-12-01	23.34
2020-12-02	23.57
2020-12-03	22.73
2020-12-04	22.87
2020-12-05	23.39
2020-12-06	24.88
2020-12-07	23.95
2020-12-08	23.91
2020-12-09	24.67
2020-12-10	25.43
2020-12-11	26.81
2020-12-12	25.36
2020-12-13	23.74
2020-12-14	24.91
2020-12-15	25.10
2020-12-16	25.92
2020-12-17	25.53
2020-12-18	26.10
2020-12-19	25.00
2020-12-20	25.06
2020-12-21	25.19
2020-12-22	26.65
2020-12-23	27.46
2020-12-24	27.19
2020-12-25	27.02
2020-12-26	27.68
2020-12-27	26.81
2020-12-28	25.81
2020-12-29	25.62
2020-12-30	26.13
2020-12-31	26.67
2021-01-01	27.03
2021-01-02	26.45
2021-01-03	27.68

Date	Forecasted Max Temperature
2021-01-04	26.70
2021-01-05	27.10
2021-01-06	25.67
2021-01-07	25.92
2021-01-08	26.06
2021-01-09	27.00
2021-01-10	26.44
2021-01-11	26.97
2021-01-12	26.03
2021-01-13	26.10
2021-01-14	27.03
2021-01-15	27.34
2021-01-16	26.94
2021-01-17	26.52
2021-01-18	27.35
2021-01-19	25.89
2021-01-20	26.30
2021-01-21	27.59
2021-01-22	26.74
2021-01-23	27.57
2021-01-24	27.73
2021-01-25	27.81
2021-01-26	28.06
2021-01-27	27.99
2021-01-28	27.97
2021-01-29	27.20
2021-01-30	26.56
2021-01-31	26.00
2021-02-01	27.30
2021-02-02	27.41
2021-02-03	27.58
2021-02-04	28.13
2021-02-05	27.59
2021-02-06	26.98
2021-02-07	28.11
2021-02-08	27.42
2021-02-09	26.32
2021-02-10	26.33
2021-02-11	26.60
2021-02-12	26.41
2021-02-13	27.41
2021-02-14	26.08
2021-02-15	25.96
2021-02-16	26.90
2021-02-17	26.24
2021-02-18	26.39
2021-02-19	26.17
2021-02-20	25.86
2021-02-21	26.50
2021-02-22	26.79
2021-02-23	26.52
2021-02-24	25.59

Date	Forecasted Max Temperature
2021-02-25	25.19
2021-02-26	26.69
2021-02-27	25.93
2021-02-28	26.04
2021-03-01	26.93
2021-03-02	27.46
2021-03-03	27.23
2021-03-04	25.11
2021-03-05	24.49
2021-03-06	24.28
2021-03-07	25.69
2021-03-08	25.36
2021-03-09	25.04
2021-03-10	25.43
2021-03-11	25.33
2021-03-12	25.04
2021-03-13	24.52
2021-03-14	24.60
2021-03-15	23.14
2021-03-16	24.41
2021-03-17	25.90
2021-03-18	24.60
2021-03-19	23.59
2021-03-20	23.00
2021-03-21	22.82
2021-03-22	22.99
2021-03-23	21.59
2021-03-24	23.56
2021-03-25	22.06
2021-03-26	22.43
2021-03-27	23.88
2021-03-28	23.84
2021-03-29	23.43
2021-03-30	22.39
2021-03-31	22.70
2021-04-01	22.21
2021-04-02	21.87
2021-04-03	22.36
2021-04-04	23.18
2021-04-05	21.67
2021-04-06	20.92
2021-04-07	21.39
2021-04-08	20.71
2021-04-09	20.36
2021-04-10	20.62

Building models to predict the number of extreme heat days